DECADE
Internet-Draft
Intended status: Informational
Expires: May 3, 2012

Y. Gu
Huawei
D. Bryan
Polycom, Inc.
Y. Yang
Yale University
R. Alimi
Google
October 31, 2011

DECADE Requirements
draft-ietf-decade-reqs-05

Abstract

The target of DECoupled Application Data Enroute (DECADE) is to provide an open and standard in-network storage system for applications, primarily P2P (peer-to-peer) applications, to store, retrieve and manage their data. This draft enumerates and explains requirements, not only for storage and retrieval, but also for data management, access control and resource control, that should be considered during the design and implementation of a DECADE system. These are requirements on the entire system; some of the requirements may eventually be implemented by an existing protocol with/without some extensions (e.g., a protocol used to read and write data from the storage system). The requirements in this document are intended to ensure that the DECADE architecture includes all of the desired functionality for intended applications.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on May 3, 2012.

Copyright Notice

Table of Contents

1. Introduction

The object of DECoupled Application Data Enroute (DECADE) is to provide an open and standard in-network storage system for content distribution applications, where data is typically broken into one or more chunks and then distributed. This may already include many types of applications including P2P applications, IPTV (Internet Protocol Television), and VoD (Video on Demand). (For a precise definition of the applications targeted in DECADE, see the definition for Target Application in Section 2.) Instead of always transferring data directly from a source/owner client to a requesting client, the source/owner client can write to and manage its content on its in-network storage. The requesting client can get the address of the in-network storage pertaining to the source/owner client and read data from the storage.

This draft enumerates and explains the rationale behind SPECIFIC requirements on the protocol design and on any data store implementation that may be used to implement DECADE servers that should be considered during the design and implementation of a DECADE system. As such, it DOES NOT include general guiding principles. General design considerations, explanation of the problem being addressed, and enumeration of the types of applications to which DECADE may be suited is not considered in this document. For general information, please see the problem statement [I-D.ietf-decade-problem-statement] and architecture [I-D.ietf-decade-arch] drafts.

This document enumerates the requirements to enable target applications to utilize in-network storage. In this context, using storage resources includes not only basic capabilities such as writing, reading, and managing data, but also controlling access for particular remote clients with which it is sharing data. Additionally, we also consider controlling the resources used by remote clients when they access data as an integral part of utilizing the network storage.

This document discusses requirements pertaining to DECADE protocol(s). In certain deployments, several logical in-network storage systems could be deployed (e.g., within the same administrative domain). These in-network storage systems can communicate and transfer data through internal or non-standard communication messages that are outside of the scope of these requirements, but they SHOULD use DECADE protocol(s) when communicating with other DECADE-capable in-network storage systems.

2. Terminology and Concepts

This document uses terms defined in [I-D.ietf-decade-problem-statement].

This document also defines additional terminology:

Target Application: An application (typically installed at end-hosts) with the ability to explicitly control usage of network and/or storage resources to deliver content to a large number of users. This includes scenarios where multiple applications or entities cooperate, such as with P2P, CDN, and hybrid P2P/CDN architectures. Such applications distribute large amounts of content (e.g., a large file, or video stream) by dividing the content into smaller blocks for more flexible distribution (e.g., over multiple application-level paths). The distributed content is typically immutable (though it may be deleted). We use the term Target Application to refer to the type of applications that are explicitly (but not exclusively) supported by DECADE.

3. Requirements Structure

The DECADE protocol is intended to sit between Target Applications and a back-end storage system. DECADE does not intend to develop yet another storage system, but rather to create a protocol that enables Target Applications to make use of storage within the network, leaving specific storage system considerations to the implementation of the DECADE servers as much as possible. For this reason, we have divided the requirements into two primary categories:

- Protocol Requirements: Protocol requirements for Target Applications to make use of in-network storage within their own data dissemination schemes. Development of these requirements is guided by a study of data access, search and management capabilities used by Target Applications. These requirements may be met by a combination of existing protocols and new protocols.

- Storage Requirements: Functional requirements necessary for the back-end storage system employed by the DECADE server. Development of these requirements is guided by a study of the data access patterns used by Target Applications. These requirements should be met by the underlying data transport used by DECADE. In this document, we use "data transport" to refer to a protocol used to read and write data from DECADE in-network storage.

Note that a third category also enumerates requirements on the protocol used to discover DECADE Servers.

It should also be made clear that the approach is to make DECADE a simple protocol, while still enabling its usage within many Target Applications. For this reason, and to further reinforce the distinction between DECADE and a storage system, in some cases we also highlight the non-requirements of the protocol. These non-requirements are intended to capture behaviors that will NOT be assumed to be needed by DECADE's Target Applications and hence not present in the DECADE protocol.

Finally, some implementation considerations are provided, which while not strictly requirements, are intended to provide guidance and highlight potential points that need to be considered by the protocol developers, and later by implementors.

4. Protocol Requirements

This section details the requirements of DECADE protocol(s).

4.1. Overall Protocol Requirements

4.1.1. Connectivity Concerns

4.1.1.1. NATs and Firewalls

REQUIREMENT(S): DECADE client to server protocols SHOULD be usable across firewalls and NAT (Network Address Translation) devices without requiring additional network support (e.g., Application-level Gateways).

RATIONALE: Firewalls and NATs are widely used in the Internet today, both in ISP (Internet Service Provider) and Enterprise networks and by consumers. Deployment of DECADE must not require modifications to such devices (beyond, perhaps, reconfiguration). Note that this requirement applies to both any new protocol developed by the DECADE Working Group and any data transport used with DECADE.

4.1.1.2. Connections to Clients

REQUIREMENT(S): DECADE SHOULD require that network connections be made from DECADE clients to DECADE servers (i.e., not from the server to the DECADE client).

RATIONALE: Many household networks and operating systems have firewalls and NATs configured by default to block incoming connections. To ease deployment by avoiding configuration changes and help mitigate security risks, DECADE should not

require clients to listen for any incoming network connections (beyond what is required by any other already-deployed application).

4.1.2. Security

4.1.2.1. Secure Transport

REQUIREMENT(S): DECADE MUST contain a mode in which all communication between a DECADE client and server is over a secure transport that provides confidentiality, integrity, and authentication.

RATIONALE: Target Applications may wish to write sensitive data. To satisfy this use case, DECADE must provide a mode in which all communication between a DECADE client and server occurs over a secure transport protocol (e.g., SSL/TLS).

4.1.3. Error and Failure Conditions

4.1.3.1. Overload Condition

REQUIREMENT(S): In-network storage, which is operating close to its capacity limit (e.g., too busy servicing other requests), MUST be permitted to reject requests and not be required to generate responses to additional requests. In-network storage MUST also be permitted to redirect requests (see Section 4.1.3.5) as a load-shedding technique.

RATIONALE: Forcing the in-network storage to respond to requests when operating close to its capacity can impair its ability to service existing requests, and thus is permitted to avoid generating responses to additional requests.

4.1.3.2. Insufficient Resources

REQUIREMENT(S): DECADE MUST support an error condition indicating to a DECADE client that resources (e.g., storage space) were not available to service a request (e.g., storage quota exceeded when attempting to write data).

RATIONALE: The currently-used resource levels within the in-network storage are not locally-discoverable, since the resources (disk, network interfaces, etc) are not directly attached. In order to allocate resources appropriately amongst remote clients, a client must be able to determine when resource limits have been reached. The client can then respond by explicitly freeing necessary resources or waiting for such resources to be freed.

4.1.3.3. Unavailable and Deleted Data

REQUIREMENT(S): DECADE MUST support error conditions indicating that (1) data was rejected from being written, (2) deleted, or (3) marked unavailable by a storage provider.

RATIONALE: Storage providers may require the ability to (1) avoid storing, (2) delete, or (3) quarantine certain data that has been identified as illegal (or otherwise prohibited). DECADE does not indicate how such data is identified, but applications using DECADE should not break if a storage provider is obligated to enforce such policies. Appropriate error conditions should be indicated to applications.

4.1.3.4. Insufficient Permissions

REQUIREMENT(S): DECADE MUST support error conditions indicating that the requesting client does not have sufficient permissions to access requested data objects.

RATIONALE: DECADE clients may request objects to which they do not have sufficent access permissions, and DECADE servers must be able to signal that this has occurred. Note that access permissions may be insufficient due to a combination of the credentials presented by a client as well as additional policies defined by the storage provider.

4.1.3.5. Redirection

REQUIREMENT(S): DECADE SHOULD support the ability for a DECADE server to redirect requests to another DECADE server. This may either be in response to an error, failure, or overload condition, or to support capabilities such as load balancing.

RATIONALE: A DECADE server may opt to redirect requests to another server to support capabilities such as load balancing, or if the implementation decides that another DECADE server is in a better position to handle the request due to either its location in the network, server status, or other consideration.

4.2. Transfer and Latency Requirements

4.2.1. Low-Latency Access

REQUIREMENT(S): DECADE SHOULD provide "low-latency" access for application clients. DECADE MUST allow clients to specify at least two classes of services: lowest possible latency and latency non-critical.

RATIONALE: Some applications may have requirements on delivery time (e.g., live streaming [PPLive]). The user experience may be unsatisfactory if the use of in-network storage results in lower performance than connecting directly to remote clients over a low-speed, possibly congested uplink. Additionally, the overhead required for control-plane operations in DECADE must not cause the latency to be higher than for a low-speed, possibly congested uplink. While it is impossible to make a guarantee that a system using in-network storage will always outperform a system that does not for every transfer, the overall performance of the system should be improved compared with direct low-speed uplink connections, even considering control overhead.

4.2.2. Data Object Size

REQUIREMENT(S): DECADE MUST allow for efficient data transfer of small objects (e.g., 16KB) between a DECADE client and in-network storage with minimal additional latency imposed by the protocol.

RATIONALE: Though Target Applications are frequently used to share large amounts of data (e.g., continuous streams or large files), the data itself is typically subdivided into smaller chunks that are transferred between clients. Additionally, the small chunks may have requirements on delivery time (e.g., in a live-streaming application). DECADE must enable data to be efficiently transferred amongst clients at this granularity. It is important to note that DECADE may be used to store and retrieve larger objects, but protocol(s) should not be designed such that usage with smaller data objects cannot meet the requirements of Target Applications.

4.2.3. Communication among DECADE Servers

REQUIREMENT(S): DECADE SHOULD support the ability for two in-network storage elements in different administrative domains to write and/or read data directly between each other (if authorized as described in Section 4.7). If such a capability is supported, this MAY be the same (or a subset or extension of) as the DECADE protocol(s) used by clients to access data.

RATIONALE: Allowing server-to-server communication can reduce latency in some common scenarios. Consider a scenario when a DECADE client is downloading data into its own storage from another client's in-network storage. One possibility is for the client to first download the data itself, and then upload it to its own storage. However, this uploading causes unnecessary latency and network traffic. Allowing the data to be downloaded from the remote in-network storage into the client's own in-network storage can alleviate both.

4.3. Data Access Requirements

4.3.1. Reading/Writing Own Storage

REQUIREMENT(S): DECADE MUST support the ability for a DECADE client to read data from and write data to its own in-network storage.

RATIONALE: Two basic capabilities for any storage system are reading and writing data. A DECADE client can read data from and write data to in-network storage space that it owns.

4.3.2. Access by Other Users

REQUIREMENT(S): DECADE MUST support the ability for a user to apply access control policies to users other than itself for its storage. The users with whom access is being shared can be under a different administrative domain than the user who owns the in-network storage. The authorized users may read from or write to the user's storage.

RATIONALE: Endpoints in Target Applications may be located across multiple ISPs under multiple administrative domains. Thus, to be useful by Target Applications, DECADE allows a user to implement access control policies for users that may or may not be known to the user's storage provider.

4.3.3. Negotiable Data Transport Protocol

REQUIREMENT(S): DECADE MUST support the ability for a DECADE client to negotiate with its in-network storage about which protocol it can use to read data from and write data to its In-network storage. DECADE MUST specify at least one mandatory protocol to be supported by implementations; usage of a different protocol may be selected via negotiation.

RATIONALE: Since typical data transport protocols (e.g., NFS and WebDAV) already provide read and write operations for network storage, it may not be necessary for DECADE to define such operations in a new protocol. However, because of the particular application requirements and deployment considerations, different applications may support different protocols. Thus, a DECADE client must be able to select an appropriate protocol also supported by the in-network storage. This requirement also follows as a result of the requirement of Separation of Control and Data Operations (Section 4.3.4).

4.3.4. Separation of Data and Control Policies

REQUIREMENT(S): DECADE Protocol(s) MUST only provide a minimal set of core operations to support diverse policies implemented and desired by Target Applications.

RATIONALE: Target Applications support many complex behaviors and diverse policies to control and distribute data, such as (e.g., search, index, setting permissions/passing authorization tokens). Thus, to support such Target Applications, these behaviors must be logically separated from the data transfer operations (e.g., read, write). Some minimal overlap (for example obtaining credentials needed to encrypt or authorize data transfer using control operations) may be required to be directly specified by DECADE.

4.4. Data Management Requirements

4.4.1. Agnostic of reliability

REQUIREMENT(S): DECADE SHOULD remain agnostic of reliability/ fault-tolerance level offered by storage provider.

RATIONALE: Providers of a DECADE service may wish to offer varying levels of service for different applications/users. However, a single compliant DECADE client should be able to use multiple DECADE services with differing levels of service.

4.4.2. Data Object Attributes

REQUIREMENT(S): DECADE MUST support the ability to associate attributes with data objects with a scope local to a DECADE server, and for DECADE clients to query these attributes. DECADE protocol(s) MUST transmit any attributes using an operating system-independent and architecture-independent standard format. DECADE protocol(s) MUST be designed such that new attributes can be added by later protocol revisions or extensions.

RATIONALE: DECADE supports associating attributes (e.g., a time-to-live, creation timestamp, or object size) with a data object. These attributes are local to a data object stored by a particular DECADE server, and are thus not applied to any DECADE server or client to which the data object is copied. These attributes may be used as hints to the storage system, internal optimizations, or as additional information queryable by DECADE clients.

4.4.3. Time-to-live for Written Data Objects

REQUIREMENT(S): DECADE MUST support the ability for a DECADE client to indicate a time-to-live value (or expiration time) indicating a length of time until particular data can be deleted by the in-network storage element.

RATIONALE: Some data objects written by a DECADE client may be usable only within a certain window of time, such as in live-streaming P2P applications. Providing a time-to-live value for data objects (e.g., at the time they are written) can reduce management overhead by avoiding many 'delete' commands sent to in-network storage. The in-network storage may still keep the data in cache for bandwidth optimization. But this is guided by the privacy policy of the DECADE provider.

4.4.4. Offline Usage

REQUIREMENT(S): DECADE MAY support the ability for a user to provide authorized access to its in-network storage even if the user has no DECADE applications actively running or connected to the network.

RATIONALE: If an application desires, it can authorize remote clients to access its storage even after the application exits or network connectivity is lost. An example use case is mobile scenarios, where a client can lose and regain network connectivity very often.

4.5. Data Naming Requirements

4.5.1. Unique Names

REQUIREMENT(S): DECADE MUST support a naming scheme that ensures a high probability of uniqueness and supports the operation of DECADE servers under diverse administrative domains with no coordination. DECADE SHOULD provide a mechanism (minimally rejection) to handle the improbable case of a collision.

RATIONALE: When writing a data object, a DECADE Client should be able to write it without being concerned over whether an object of the same name already exists, unless the existing object contains the exact same data. Note that it may be reasonable for DECADE to satisfy this requirement probabilistically (e.g., using a hashing mechanism). As a result, it is wise to provide a collision handling mechanism, even if the probability of collisions is extremely low.

4.6. Resource Control

4.6.1. Multiple Applications

REQUIREMENT(S): DECADE SHOULD support the ability for users to define resource sharing policies for multiple applications (DECADE clients) being run/managed by the user.

RATIONALE: A user may own in-network storage and share the in-network storage resources amongst multiple applications. For example, the user may run one or more video-on-demand application(s) and a live-streaming application(s) which both make use of the user's in-network storage. The applications may be running on different machines and may not directly communicate. Thus, DECADE should enable the user to determine resource sharing policies between the applications.

One possibility is for a user to indicate the particular resource sharing policies between applications out-of-band (not using a standard protocol), but this requirement may manifest itself in passing values within DECADE protocol(s) to identify individual applications. Such identifiers can be either user-generated or server-generated and do not need to be registered by IANA.

4.6.2. Per-Remote-Client, Per-Data Control

REQUIREMENT(S): A DECADE client MUST be able to assign resource policies (bandwidth share, storage quota, and network connection quota) to individual remote clients for reading from and writing particular data to its in-network storage within a particular range of time. The DECADE server MUST enforce these constraints.

RATIONALE: Target Applications can rely on control of resources on a per-remote-client or per-data basis. For example, application policy may indicate that certain remote clients have a higher bandwidth share for receiving data [LLSB08]. Additionally, certain data (e.g., chunks) may be distributed with a higher priority. As another example, when allowing a remote client to write data to a user's in-network storage, the remote client may

be restricted to write only a certain amount of data. Since the client may need to manage multiple clients accessing its data, it should be able to indicate the time over which the granted resources are usable. For example, an expiration time for the access could be indicated to the server after which no resources are granted (e.g., indicate error as access denied).

4.6.3. Server Involvement

REQUIREMENT(S): A DECADE client SHOULD be able to indicate to a DECADE server, without itself contacting the server, resource control policies for remote clients' requests.

RATIONALE: One important consideration for in-network storage elements is scalability, since a single storage element may be used to support many users. Many Target Applications use small chunk sizes and frequent data exchanges. If such an application employed resource control and contacted the in-network storage element for each data exchange, this could present a scalability challenge for the server as well as additional latency for clients.

Our preferred alternative is to let requesting users obtain signed resource control policies (in the form of a token) from the owning user, and then users can then present the policy to the storage directly. This can result in reduced messaging handled by the in-network storage.

4.7. Authorization

4.7.1. Per-Remote-Client, Per-Data Read Access

REQUIREMENT(S): A DECADE Client MUST be able to control which individual remote clients are authorized to read particular data from its in-network storage.

RATIONALE: A Target Application can control certain application-level policies by sending particular data (e.g., chunks) to certain remote clients. It is important that remote clients not be able to circumvent such decisions by arbitrarily reading any data in in-network storage.

4.7.2. Per-User Write Access

REQUIREMENT(S): A DECADE Client MUST be able to control which individual remote clients are authorized to write data into its in-network storage.

RATIONALE: The space managed by a user in in-network storage can be a limited resource. At the same time, it can be useful to allow remote clients to write data directly to a user's in-network storage. Thus, a DECADE client should be able to grant only certain remote clients this privilege.

4.7.3. Default Access Permissions

REQUIREMENT(S): Unless read or write access is granted by a DECADE Client to a remote client, the default access MUST be no access.

RATIONALE: The current leading proposal for an access control model in DECADE is via token passing, resulting in a system with little state on the server side.

4.7.4. Authorization Checks

REQUIREMENT(S): In-network storage MUST check the authorization of a client before it executes a supplied request. The in-network storage MAY use optimizations to avoid such authorization checks as long as the enforced permissions are the same.

RATIONALE: Authorization granted by a DECADE client are meaningless unless unauthorized requests are denied access. Thus, the in-network storage element must verify the authorization of a particular request before it is executed.

4.7.5. Cryptographic Credentials

REQUIREMENT(S): Access MUST be able to be granted using cryptographic credentials.

RATIONALE: DECADE clients may be operating on hosts without constant network connectivity or without a permanent attachment address (e.g., mobile devices). To support access control with such hosts, DECADE servers must support access control policies that use cryptographic credentials, not simply by tying access to IP addresses.

4.7.6. Server Involvement

REQUIREMENT(S): A DECADE client SHOULD be able to indicate, without contacting the server itself, access control policies for remote clients' requests.

RATIONALE: See discussion in Section 4.6.3.

4.7.7. Protocol Reuse

REQUIREMENT(S): If possible, DECADE SHOULD reuse existing protocol and mechanisms for Authentication, Access, and Authorization (AAA).

RATIONALE: If possible, reusing an existing AAA protocol/mechanism will minimize the development required by applications, and will maximize interoperability of the DECADE protocol with existing infrastructure.

4.8. Non-Requirements

4.8.1. Application-defined Properties and Metadata

REQUIREMENT(S): DECADE MUST NOT provide a mechanism for associating Application-defined properties (metadata) with data objects beyond what is provided by Section 4.4.2.

RATIONALE: Associating key-value pairs that are defined by Target Applications with data objects introduces substantial complexity to the DECADE protocol. If Target Applications wish to associate additional metadata with a data object, possible alternatives include (1) managing such metadata within the Target Application itself, (2) storing metadata inside of the data object, or (3) storing metadata in a different data object at the DECADE server.

5. Storage Requirements

This section details the requirements of the underlying storage used to support the DECADE protocol(s).

5.1. Immutable Data

REQUIREMENT(S): DECADE MUST only store and manage data objects that are immutable once they are written to storage.

RATIONALE: Immutable data objects are an important simplification in DECADE. Reasonable consistency models for updating existing objects would significantly complicate implementation (especially if implementation chooses to replicate data across multiple

servers). If a user needs to update a resource, it can write a new resource and then distribute the new resource instead of the old one.

5.2. Explicit Deletion of Data

REQUIREMENT(S): DECADE MUST support the ability for a DECADE client to explicitly delete data from its own in-network storage.

RATIONALE: A DECADE client may continually be writing data to its in-network storage. Since there may be a limit (e.g., imposed by the storage provider) to how much total storage can be used, some data may need to be removed to make room for additional data. A DECADE client should be able to explicitly remove particular data. This may be implemented using existing protocols.

5.3. Multiple writing

REQUIREMENT(S): DECADE MUST NOT allow multiple simultaneous writers for the same object. Implementations MUST raise an error to one of the writers.

RATIONALE: This avoids data corruption in a simple way while remaining efficient. Alternately, the DECADE server would need to either manage locking, handle write collisions, or merge data, all of which reduce efficiency and increase complexity.

5.4. Multiple reading

REQUIREMENT(S): DECADE MUST allow for multiple simultaneous readers for an object.

RATIONALE: One characteristic of Target Applications is the ability to upload an object to multiple clients. Thus, it is natural for DECADE to allow multiple readers to read the content concurrently.

5.5. Reading before completely written

REQUIREMENT(S): DECADE MAY allow readers to read from objects before they have been completely written.

RATIONALE: Some Target Applications (in particular, P2P streaming) can be sensitive to latency. A technique to reduce latency is to remove store-and-forward delays for data objects (e.g., make the object available before it is completely written). Appropriate handling for error conditions (e.g., a disappearing writer) needs to be specified.

5.6. Hints concerning usage of written data

REQUIREMENT(S): DECADE MAY allow writers of new objects to indicate specific hints concerning how the objects are expected to be used (e.g., access frequency or time-to-live).

RATIONALE: Different Target Applications may have different usage patterns for objects written to in-network storage. For example, a P2P live streaming application may indicate to a DECADE server that the objects are expected to have a short time-to-live, but read frequently. The DECADE server may then opt to persist the objects in memory instead of in disk.

5.7. Writing model

REQUIREMENT(S): DECADE storage MUST provide at least a writing model (while writing an object) that appends data to data already written.

RATIONALE: Depending on the object size (e.g., chunk size) used by a Target Application, the application may need to send data to the DECADE server in multiple packets. To keep implementation simple, the DECADE must at least support the ability to write the data sequentially in the order received. Implementations MAY allow application to write data in an object out-of-order (but MUST NOT overwrite ranges of the object that have already been written).

5.8. Storage Status

REQUIREMENT(S): A DECADE client MUST be able to read current resource usage (including list of written data objects), resource quotas, and access permissions for its in-network storage. The returned information MUST include resource usage resulting from the client's own usage and usage by other clients that have been authorized to read/write objects or open connections to that client's storage. DECADE protocol(s) MUST support the ability for a DECADE client to read aggregated resource usage information (across all other clients to which it has authorized access), and MAY support the ability to request this information for each other authorized client.

RATIONALE: The resources used by a client are not directly-attached (e.g., disk, network interface, etc). Thus, the client cannot locally determine how such resources are being used. Before storing and retrieving data, a client should be able to determine which data is available (e.g., after an application restart). Additionally, a client should be able to determine resource

availability to better allocate them to remote clients. Due to scalability issues, it is not required that DECADE support returning usage information broken down by each remote client which has been authorized access, but this feature may be useful in certain deployment scenarios.

6. Discovery Requirements

6.1. Requirements

6.1.1. Locating DECADE Servers

REQUIREMENT(S): The DECADE Discovery mechanism MUST allow a DECADE Client to identify one or more DECADE Servers to which it is authorized to read/write data and to which it may authorize other DECADE Clients to read/write data, or fail if no such DECADE Servers are available.

RATIONALE: A basic goal of DECADE is to allow DECADE Clients to read/write data for access by other DECADE Clients or itself. Executing the Discovery process should result in a DECADE Client finding a DECADE Server that it can use for these purposes. It is possible that no such DECADE Servers are available. Note that even if a DECADE Client may not successfully locate a DECADE Server that fits these criteria, it may still read/write data from/to a DECADE Server if authorized by another DECADE Client.

6.1.2. Support for Clients Behind NATs and Firewalls

REQUIREMENT(S): The Discovery mechanism MUST support DECADE Clients operating behind NATs and Firewalls without requiring additional network support (e.g., Application-level Gateways).

RATIONALE: NATs and Firewalls are prevalent in network deployments, and it is common for Target Applications that include a DECADE Client to be deployed behind these devices.

6.1.3. Prefer Existing Protocols

REQUIREMENT(S): The DECADE Server discovery mechanism SHOULD leverage existing mechanisms and protocols wherever possible.

RATIONALE: Existing protocols such as DNS and DHCP are widespread. Using existing protocols, or combinations of the protocols that have been specified in other contexts, is strictly preferred over developing a new discovery protocol for DECADE.

7. Future Considerations

This section enumerates considerations that should be taken into account during the DECADE design and implementation. They have been intentionally omitted as requirements since they are either out of scope or implementation-dependent. Nevertheless, enumerating them may help to guide future application of the requirements included in this document.

7.1. Fairness

To provide fairness among users, the in-network storage provider should assign resource (e.g., storage, bandwidth, connections) quota for users. This can prevent a small number of clients from occupying large amounts of resources on the in-network storage, while others starve.

7.2. Removal of Duplicate Data Objects

There are actually two possible scenarios. The first is the case of removing duplicates within one particular DECADE server (or logical server). While not a requirement, as it does not impact the protocol, a DECADE server may implement internal mechanisms to monitor for duplicate objects and use internal mechanisms to prevent duplication in internal storage.

The second scenario is removing duplicates across a distributed set of DECADE servers. DECADE does not explicitly design for this, but does offer a redirection mechanism (Section 4.1.3.5) that is one primitive that may be used to support this feature in certain deployment scenarios.

7.3. Gaming of the Resource Control Mechanism

The particular resource control policy communicated by a DECADE protocol and enforced by the scheduling system of a DECADE implementation may be open to certain gaming by clients. for example by specifying many small peers to increase total throughput or inciting overload conditions at a DECADE server. Identifying and protecting against all such opportunities for gaming is outside the scope of this document, but DECADE protocol(s) and implementations SHOULD be aware that opportunities to game the system may be attempted.

8. Security Considerations

The security model is an important component of DECADE. It is

crucial for users to be able to manage and limit distribution of content to only authorized parties, and the mechanism needs to work on the general Internet which spans multiple administrative and security domains. Previous sections have enumerated detailed requirements, but this section discusses the overall approach and other considerations that do not merit requirements.

8.1. Authentication and Authorization

DECADE only uses authentication when allowing a particular client to access its own storage at a server. DECADE servers themselves do not authenticate other clients which are reading/writing a client's own storage. Instead, DECADE relies on clients to authenticate others to access its storage, and then communicate the result of that authentication to the DECADE server so that the DECADE server may implement the necessary authorization checks.

8.2. Encrypted Data

DECADE Servers provide the ability to write raw data objects (subject to any policies instituted by the owner/administrator of the DECADE server, which are outside of the scope of this document). Thus, DECADE clients may opt to encrypt data before it is written to the DECADE Server. However, DECADE itself does not provide encryption of data objects other than is provided by Section 4.1.2.1.

9. IANA Considerations

There are no IANA considerations with this document.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

10.2. Informative References

[I-D.ietf-decade-problem-statement]
Song, H., Zong, N., Yang, Y., and R. Alimi, "DECoupled Application Data Enroute (DECADE) Problem Statement", draft-ietf-decade-problem-statement-03 (work in progress), March 2011.

[I-D.ietf-decade-arch]

Alimi, R., Yang, Y., Rahman, A., Kutscher, D., and H. Liu, "DECADE Architecture", draft-ietf-decade-arch-02 (work in progress), July 2011.

[LLSB08] Levin, D., LaCurts, K., Spring, N., and B. Bhattacharjee, "BitTorrent is an Auction: Analyzing and Improving BitTorrent's Incentives", SIGCOMM 2008, August 2008.

[PPLive] "PPLive", <http://www.pplive.com>.

Appendix A. Acknowledgments

We would also like to thank Haibin Song for substantial contributions to earlier versions of this document. We would also like to thank Reinaldo Penno, Alexey Melnikov, Rich Woundy, Ning Zong, Roni Even, David McDysan, Borje Ohlman, Dirk Kutscher, Akbar Rahman, Xiao Zhu, Yunfei Zhang, and Jin Peng for contributions and general feedback.

Authors' Addresses

Yingjie Gu
Huawei
No. 101 Software Avenue
Nanjing, Jiangsu Province 210012
P.R.China

Phone: +86-25-56624760
Email: guyingjie@huawei.com

David A. Bryan
Polycom, Inc.

Email: dbryan@ethernot.org

Yang Richard Yang
Yale University

Email: yry@cs.yale.edu

Richard Alimi
Google

Email: ralimi@google.com